Hexaware

Assignment 1

Task:1. Database Design:

1. **Create the database named "TechShop"**
2. **Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.**
3. **Create appropriate Primary Key and Foreign Key constraints for referential integrity.**

```
mysql> create database TechShop;
Query OK, 1 row affected (0.05 sec)
```

```
mysql> use TechShop;
Database changed
mysql> create table Customers(CustomerID int,FirstName text, LastName text,Email varchar(20),Pho
ne int,Address text);
Query OK, 0 rows affected (0.10 sec)

mysql> create table Products(ProductID int,ProductName text, Description text,Price int);
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Orders (
    ->     OrderID INT PRIMARY KEY,
    ->     CustomerID INT,
    ->     OrderDate DATE,
    ->     TotalAmount DECIMAL(10, 2),
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
    -> );
```

```
mysql> CREATE TABLE OrderDetails(
    -> OrderDetailID INT PRIMARY KEY,
    -> OrderID INT,
    -> ProductID INT,
    -> Qunatity INT,
    -> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
```

```
mysql> CREATE TABLE Inventory (
    ->     InventoryID INT PRIMARY KEY,
    ->     ProductID INT,
    ->     QuantityInStock INT,
    ->     LastStockUpdate DATE,
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
```

4. **Insert at least 10 sample records into each of the following tables.**
   a. Customers
   b. Products
   c. Orders
   d. OrderDetails

```
mysql> select * from Customers;
+------------+-----------+----------+---------------------+------------+-----------+
| CustomerID | FirstName | LastName | Email               | Phone      | Address   |
+------------+-----------+----------+---------------------+------------+-----------+
|        101 | Ram       | Goud     | ramgoud1@gmail.com  | 9876543210 | Bengaluru |
|        102 | John      | Smith    | johnsmith1@gmail.com| 9876543211 | Bidar     |
|        103 | Roshan    | Kumar    | roshkumar1@gmail.com| 9740116621 | Belgaum   |
|        104 | Shamanth  | Gouda    | shamgoud1@gmail.com | 9876543213 | Ballari   |
|        105 | Suchetha  | Patil    | suchpatil1@gmail.com| 8762530197 | Davangere |
|        106 | Prathibha | Patil    | pratipatil@gmail.com| 9632657540 | Durga     |
|        107 | Sam       | john     | samjoy@gmail.com    | 9636821401 | Harihar   |
|        108 | John      | Brew     | ajay123@gmail.com   | 9636821402 | Gokak     |
|        109 | Robert    | Downy    | john123@gmail.com   | 9636821403 | Gokarna   |
|        110 | Samson    | Rock     | samrok123@gmail.com | 9636821405 | Coorg     |
+-----------+-----------+----------+---------------------+------------+-----------+
10 rows in set (0.00 sec)
```

```
mysql> select * from Products;
+-----------+-------------+-----------------------------------+--------+
| ProductID | ProductName | Description                       | Price  |
+-----------+-------------+-----------------------------------+--------+
|         1 | Laptop      | Brand-Dell, Ram-64 bit-processor  |  45000 |
|         2 | Mobile      | Brand-Redmi, snapdragon processor |  50000 |
|         3 | Tab         | Brand-Realme, snapdragon processor|  72000 |
|         4 | System      | Brand-LG, Ram-8                   |  56000 |
|         5 | VR          | Brand-X, Speed                    | 125000 |
|         6 | AR          | Brand-Y, High Speed               | 175000 |
|         7 | Watch       | Brand-Fastrack, smart watch       |   5000 |
|         8 | Earphones   | Brand-Boat, Bluetooth earphones   |   1500 |
|         9 | TV          | Brand-Samsung, Smart TV           |  50000 |
|        10 | Mac         | Brand-Apple, smart                | 125000 |
+-----------+-------------+-----------------------------------+--------+
10 rows in set (0.06 sec)
```

```
mysql> select * from Orders2;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |        101 | 2020-06-15 |    42000.00 |
|       2 |        102 | 2020-09-25 |    45000.00 |
|       3 |        103 | 2021-10-20 |    55000.00 |
|       4 |        104 | 2018-12-01 |    32000.00 |
|       5 |        105 | 2023-09-23 |    15000.00 |
|       6 |        106 | 2023-06-26 |    31000.00 |
|       7 |        107 | 2022-08-12 |    45000.00 |
|       8 |        108 | 2022-12-16 |    12000.00 |
|       9 |        109 | 2022-01-18 |    91000.00 |
|      10 |        110 | 2023-02-19 |    51000.00 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

```
mysql> select *from OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Qunatity |
+---------------+---------+-----------+----------+
|             1 |       1 |         1 |        1 |
|             2 |       1 |         3 |        1 |
|             3 |       4 |         1 |        1 |
|             4 |       2 |         1 |        5 |
|             5 |       7 |         9 |        1 |
|             6 |      10 |         7 |        1 |
|             7 |       5 |         2 |        1 |
|             8 |      10 |         9 |        1 |
|             9 |       2 |         3 |        4 |
|            10 |       7 |        10 |        1 |
+---------------+---------+-----------+----------+
10 rows in set (0.00 sec)
```

```
mysql> select * from Inventory;
+-------------+-----------+-----------------+-----------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-------------+-----------+-----------------+-----------------+
|           1 |         1 |              10 | 2023-10-12      |
|           2 |         2 |              15 | 2023-01-02      |
|           3 |         3 |               3 | 2022-12-09      |
|           4 |         4 |               1 | 2023-10-05      |
|           5 |         5 |               1 | 2022-11-20      |
|           6 |         6 |               9 | 2023-02-09      |
|           7 |         7 |               7 | 2023-03-10      |
|           8 |         8 |               2 | 2022-10-19      |
|           9 |         9 |               9 | 2023-05-04      |
|          10 |        10 |              10 | 2023-10-10      |
+-------------+-----------+-----------------+-----------------+
10 rows in set (0.00 sec)
```

Task 2:

1. **Write an SQL query to retrieve the names and emails of all customers.**

```
mysql> select FirstName, LastName, Email from Customers;
+-----------+----------+----------------------+
| FirstName | LastName | Email                |
+-----------+----------+----------------------+
| Ram       | Goud     | ramgoud1@gmail.com   |
| John      | Smith    | johnsmith1@gmail.com |
| Roshan    | Kumar    | roshkumar1@gmail.com |
| Shamanth  | Gouda    | shamgoud1@gmail.com  |
| Suchetha  | Patil    | suchpatil1@gmail.com |
| Prathibha | Patil    | pratipatil@gmail.com |
| Sam       | john     | samjoy@gmail.com     |
| John      | Brew     | ajay123@gmail.com    |
| Robert    | Downy    | john123@gmail.com    |
| Samson    | Rock     | samrok123@gmail.com  |
+-----------+----------+----------------------+
10 rows in set (0.00 sec)
```

2. **Write an SQL query to list all orders with their order dates and corresponding customer names.**

```
mysql> SELECT Orders2.OrderID, Orders2.OrderDate, Customers.FirstName,Customers.LastName
    -> FROM Orders2
    -> JOIN Customers ON Orders2.CustomerID = Customers.CustomerID;
+---------+-------------+-----------+----------+
| OrderID | OrderDate   | FirstName | LastName |
+---------+-------------+-----------+----------+
|       1 | 2020-06-15 | Ram       | Goud     |
|       2 | 2020-09-25 | John      | Smith    |
|       3 | 2021-10-20 | Roshan    | Kumar    |
|       4 | 2018-12-01 | Shamanth  | Gouda    |
|       5 | 2023-09-23 | Suchetha  | Patil    |
|       6 | 2023-06-26 | Prathibha | Patil    |
|       7 | 2022-08-12 | Sam       | john     |
|       8 | 2022-12-16 | John      | Brew     |
|       9 | 2022-01-18 | Robert    | Downy    |
|      10 | 2023-02-19 | Samson    | Rock     |
+---------+-------------+-----------+----------+
10 rows in set (0.00 sec)
```

3. **Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email and address.**

```
mysql> insert into Customers(CustomerID, FirstName, LastName, Email,Address) values(11,'Harshad','Malhotra','hars@gmail.com','
Bagalkot');
Query OK, 1 row affected (0.01 sec)

mysql> select * from Customers;
+------------+-----------+----------+---------------------+------------+-----------+
| CustomerID | FirstName | LastName | Email               | Phone      | Address   |
+------------+-----------+----------+---------------------+------------+-----------+
|         11 | Harshad   | Malhotra | hars@gmail.com      | NULL       | Bagalkot  |
|        101 | Ram       | Goud     | ramgoud1@gmail.com  | 9876543210 | Bengaluru |
|        102 | John      | Smith    | johnsmith1@gmail.com | 9876543211 | Bidar     |
|        103 | Roshan    | Kumar    | roshkumar1@gmail.com | 9740116621 | Belgaum   |
|        104 | Shamanth  | Gouda    | shamgoud1@gmail.com | 9876543213 | Ballari   |
|        105 | Suchetha  | Patil    | suchpatil1@gmail.com | 8762530197 | Davangere |
|        106 | Prathibha | Patil    | pratipatil@gmail.com | 9632657540 | Durga     |
|        107 | Sam       | john     | samjoy@gmail.com    | 9636821401 | Harihar   |
|        108 | John      | Brew     | ajay123@gmail.com   | 9636821402 | Gokak     |
|        109 | Robert    | Downy    | john123@gmail.com   | 9636821403 | Gokarna   |
|        110 | Samson    | Rock     | samrok123@gmail.com | 9636821405 | Coorg     |
+------------+-----------+----------+---------------------+------------+-----------+
11 rows in set (0.00 sec)
```

4. **Write an SQL query to update the prices of all electronics gadgets in the "Products" table by increasing them by 10%.**

```
mysql> update Products set Price=Price*1.1;
Query OK, 10 rows affected (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from Products;
+-----------+-------------+--------------------------------------+--------+
| ProductID | ProductName | Description                          | Price  |
+-----------+-------------+--------------------------------------+--------+
|         1 | Laptop      | Brand-Dell, Ram-64 bit-processor     |  49500 |
|         2 | Mobile      | Brand-Redmi, snapdragon processor    |  55000 |
|         3 | Tab         | Brand-Realme, snapdragon processor   |  79200 |
|         4 | System      | Brand-LG, Ram-8                      |  61600 |
|         5 | VR          | Brand-X, Speed                       | 137500 |
|         6 | AR          | Brand-Y, High Speed                  | 192500 |
|         7 | Watch       | Brand-Fastrack, smart watch          |   5500 |
|         8 | Earphones   | Brand-Boat, Bluetooth earphones      |   1650 |
|         9 | TV          | Brand-Samsung, Smart TV              |  55000 |
|        10 | Mac         | Brand-Apple, smart                   | 137500 |
+-----------+-------------+--------------------------------------+--------+
10 rows in set (0.00 sec)

mysql>
```

5. **Write an SQL query to delete a specific order and its associated order details from the “Orders” and “OrderDetails” table. Allow users to input the order ID as a parameter.**

```
mysql> set @OrderIDToDelete=5;
Query OK, 0 rows affected (0.01 sec)

mysql> delete from OrderDetails
    -> where OrderID=@OrderIDToDelete;
Query OK, 1 row affected (0.01 sec)

mysql> delete from Orders2
    -> where OrderID=@OrderIDToDelete;
Query OK, 1 row affected (0.01 sec)
```

6. **Write an SQL query to insert a new order into the “Orders” table. Include the customers ID, order date, and any other necessary information.**

```
mysql> insert into Orders2(OrderID,CustomerID,OrderDate,TotalAmount) values(11,11,'2024-12-07',68000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Orders2;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |        101 | 2020-06-15 |    42000.00 |
|       2 |        102 | 2020-09-25 |    45000.00 |
|       3 |        103 | 2021-10-20 |    55000.00 |
|       4 |        104 | 2018-12-01 |    32000.00 |
|       6 |        106 | 2023-06-26 |    31000.00 |
|       7 |        107 | 2022-08-12 |    45000.00 |
|       8 |        108 | 2022-12-16 |    12000.00 |
|       9 |        109 | 2022-01-18 |    91000.00 |
|      10 |        110 | 2023-02-19 |    51000.00 |
|      11 |         11 | 2024-12-07 |    68000.00 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

7. **Write an SQL query to update the contact information (eg, email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.**

```
mysql> update Customers
    -> set Email='ajay124@gmail.com',Address='Gokak'
    -> where CustomerID=@CustomerID;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Customers;
+------------+-----------+----------+----------------------+------------+-----------+
| CustomerID | FirstName | LastName | Email                | Phone      | Address   |
+------------+-----------+----------+----------------------+------------+-----------+
|         11 | Harshad   | Malhotra | hars@gmail.com       | NULL       | Bagalkot  |
|        101 | Ram       | Goud     | ramgoud1@gmail.com   | 9876543210 | Bengaluru |
|        102 | John      | Smith    | johnsmith1@gmail.com | 9876543211 | Bidar     |
|        103 | Roshan    | Kumar    | roshkumar1@gmail.com | 9740116621 | Belgaum   |
|        104 | Shamanth  | Gouda    | shamgoud1@gmail.com  | 9876543213 | Ballari   |
|        105 | Suchetha  | Patil    | suchpatil1@gmail.com | 8762530197 | Davangere |
|        106 | Prathibha | Patil    | pratipatil@gmail.com | 9632657540 | Durga     |
|        107 | Sam       | john     | samjoy@gmail.com     | 9636821401 | Harihar   |
|        108 | John      | Brew     | ajay124@gmail.com    | 9636821402 | Gokak     |
|        109 | Robert    | Downy    | john123@gmail.com    | 9636821403 | Gokarna   |
|        110 | Samson    | Rock     | samrok123@gmail.com  | 9636821405 | Coorg     |
+------------+-----------+----------+----------------------+------------+-----------+
11 rows in set (0.00 sec)
```

8. **Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.**

```
mysql> update Orders
    -> set TotalAmount=(
    -> select sum(Products.Price * OrderDetails.Qunatity)
    -> from OrderDetails
    -> join Products on OrderDetails.ProductID=Products.ProductID
    -> where OrderDetails.OrderID=Orders.OrderID
    -> );
Query OK, 10 rows affected (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select *from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2020-06-15 |   105600.00 |
|       2 |          2 | 2020-09-25 |   466400.00 |
|       3 |          3 | 2021-10-20 |        NULL |
|       4 |          4 | 2018-12-01 |    44000.00 |
|       5 |          5 | 2023-12-09 |        NULL |
|       6 |          6 | 2023-06-26 |        NULL |
|       7 |          7 | 2022-08-12 |   154000.00 |
|       8 |          8 | 2022-12-16 |        NULL |
|       9 |          9 | 2022-01-18 |        NULL |
|      10 |         10 | 2023-02-19 |   148500.00 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

9. **Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.**

```
mysql> set @customerID =5;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from OrderDetails where OrderID in(
    -> select OrderID from Orders where CustomerID=@customerID);
Query OK, 0 rows affected (0.00 sec)

mysql> delete from Orders where CustomerID=@customerID;
Query OK, 1 row affected (0.01 sec)

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2020-06-15 |   105600.00 |
|       2 |          2 | 2020-09-25 |   466400.00 |
|       3 |          3 | 2021-10-20 |        NULL |
|       4 |          4 | 2018-12-01 |    44000.00 |
|       6 |          6 | 2023-06-26 |        NULL |
|       7 |          7 | 2022-08-12 |   154000.00 |
|       8 |          8 | 2022-12-16 |        NULL |
|       9 |          9 | 2022-01-18 |        NULL |
|      10 |         10 | 2023-02-19 |   148500.00 |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)
```

10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.

```
mysql> insert into Products(ProductID,ProductName,Description,Price)
    -> values(11,'Iphone','14 pro plus',90000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Products;
+-----------+-------------+-------------------------------------+--------+
| ProductID | ProductName | Description                         | Price  |
+-----------+-------------+-------------------------------------+--------+
|         1 | Laptop      | Brand-Dell, Ram-64 bit-processor    |  49500 |
|         2 | Mobile      | Brand-Redmi, snapdragon processor   |  55000 |
|         3 | Tab         | Brand-Realme, snapdragon processor  |  79200 |
|         4 | System      | Brand-LG, Ram-8                     |  61600 |
|         5 | VR          | Brand-X, Speed                      | 137500 |
|         6 | AR          | Brand-Y, High Speed                 | 192500 |
|         7 | Watch       | Brand-Fastrack, smart watch         |   5500 |
|         8 | Earphones   | Brand-Boat, Bluetooth earphones     |   1650 |
|         9 | TV          | Brand-Samsung, Smart TV             |  55000 |
|        10 | Mac         | Brand-Apple, smart                  | 137500 |
|        11 | Iphone      | 14 pro plus                         |  90000 |
+-----------+-------------+-------------------------------------+--------+
11 rows in set (0.00 sec)
```

11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

```
mysql> alter table Orders
    -> add column OrderStatus varchar(255);
Query OK, 0 rows affected (0.18 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from Orders;
+---------+------------+------------+-------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount | OrderStatus |
+---------+------------+------------+-------------+-------------+
|       1 |          1 | 2020-06-15 |   105600.00 | NULL        |
|       2 |          2 | 2020-09-25 |   466400.00 | NULL        |
|       3 |          3 | 2021-10-20 |        NULL | NULL        |
|       4 |          4 | 2018-12-01 |    44000.00 | NULL        |
|       6 |          6 | 2023-06-26 |        NULL | NULL        |
|       7 |          7 | 2022-08-12 |   154000.00 | NULL        |
|       8 |          8 | 2022-12-16 |        NULL | NULL        |
|       9 |          9 | 2022-01-18 |        NULL | NULL        |
|      10 |         10 | 2023-02-19 |   148500.00 | NULL        |
+---------+------------+------------+-------------+-------------+
9 rows in set (0.00 sec)

mysql> update Orders set OrderStatus="Shipped" where OrderID=6;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Orders;
+---------+------------+------------+-------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount | OrderStatus |
+---------+------------+------------+-------------+-------------+
|       1 |          1 | 2020-06-15 |   105600.00 | NULL        |
|       2 |          2 | 2020-09-25 |   466400.00 | NULL        |
|       3 |          3 | 2021-10-20 |        NULL | NULL        |
|       4 |          4 | 2018-12-01 |    44000.00 | NULL        |
|       6 |          6 | 2023-06-26 |        NULL | Shipped     |
|       7 |          7 | 2022-08-12 |   154000.00 | NULL        |
|       8 |          8 | 2022-12-16 |        NULL | NULL        |
|       9 |          9 | 2022-01-18 |        NULL | NULL        |
|      10 |         10 | 2023-02-19 |   148500.00 | NULL        |
+---------+------------+------------+-------------+-------------+
9 rows in set (0.00 sec)
```

12. Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.

```
mysql> UPDATE Customers
    -> SET TotalOrders = (
    ->     SELECT COUNT(*)
    ->     FROM Orders
    ->     WHERE Orders.CustomerID = Customers.CustomerID
    -> );
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> select * from Customers;
+------------+-----------+----------+--------------------------+------------+--------------------+-------------+
| CustomerID | FirstName | LastName | Email                    | Phone      | Address            | TotalOrders |
+------------+-----------+----------+--------------------------+------------+--------------------+-------------+
|          1 | Krishna   | Patle    | krishnapatle128@gmail.com | 9325654953 | 100 Shivam Nagar   |           1 |
|          2 | Kashyap   | Punyawan | kpunyawan@gmail.com       | 8698456732 | 12 Ambe Nagar      |           1 |
|          3 | Harshal   | Meshram  | harshalmeshram@gmail.com  | 9115432565 | Durga Nagar        |           1 |
|          4 | Shivam    | Kale     | shivamkale12@gmail.com    | 8765432106 | 123 Manewada       |           1 |
|          5 | Vikas     | Nagpure  | vikasnagpure@gmail.com    | 7865943215 | Ambe Nagar         |           0 |
|          6 | Nitin     | Turkar   | nitinturkar@gmail.com     | 8698454795 | Gruhalaxmi Society |           1 |
|          7 | Ruchika   | Chafekar | ruchikachafekar@gmail.com | 7447509656 | Saoner             |           1 |
|          8 | Neha      | Patle    | nehapatle@gmail.com       | 9823485622 | Shivam Nagar       |           1 |
|          9 | Pratiksha | Katre    | pratikshakatre@gmail.com  | 8475636214 | Nageshwar Nagar    |           1 |
|         10 | Shrutika  | Kolhe    | shrutkolhe@gmail.com      | 8645281931 | Saoner             |           1 |
|         11 | Masud     | Ansari   | masudansari@gmail.com     | NULL       | 87 Digori          |           0 |
+------------+-----------+----------+--------------------------+------------+--------------------+-------------+
11 rows in set (0.00 sec)
```

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. **Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.**

```
mysql> SELECT Orders2.OrderID, Orders2.OrderDate, Customers.FirstName, Customer
s.LastName
    -> FROM Orders2
    -> JOIN Customers ON Orders2.CustomerID = Customers.CustomerID;
+---------+-----------+-----------+----------+
| OrderID | OrderDate  | FirstName | LastName |
+---------+-----------+-----------+----------+
|      11 | 2024-12-07 | Harshad   | Malhotra |
|       1 | 2020-06-15 | Ram       | Goud     |
|       2 | 2020-09-25 | John      | Smith    |
|       3 | 2021-10-20 | Roshan    | Kumar    |
|       4 | 2018-12-01 | Shamanth  | Gouda    |
|       6 | 2023-06-26 | Prathibha | Patil    |
|       7 | 2022-08-12 | Sam       | john     |
|       8 | 2022-12-16 | John      | Brew     |
|       9 | 2022-01-18 | Robert    | Downy    |
|      10 | 2023-02-19 | Samson    | Rock     |
+---------+-----------+-----------+----------+
10 rows in set (0.06 sec)
```

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

```
mysql> SELECT
    ->     P.ProductName,
    ->     SUM(OD.Quantity * P.Price) AS TotalRevenue
    -> FROM
    ->     OrderDetails OD
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    ->
    -> GROUP BY
    ->     P.ProductName;
+-------------+--------------+
| ProductName | TotalRevenue |
+-------------+--------------+
| Laptop      |       346500 |
| Tab         |       396000 |
| TV          |       110000 |
| Watch       |         5500 |
| Mac         |       137500 |
+-------------+--------------+
5 rows in set (0.02 sec)
```

3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.

```
mysql> SELECT DISTINCT
    ->     C.FirstName,
    ->     C.LastName,
    ->     C.Email,
    ->     C.Phone,
    ->     C.Address
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID;
+-----------+----------+----------------------+------------+-----------+
| FirstName | LastName | Email                | Phone      | Address   |
+-----------+----------+----------------------+------------+-----------+
| Harshad   | Malhotra | hars@gmail.com       | NULL       | Bagalkot  |
| Ram       | Goud     | ramgoud1@gmail.com   | 9876543210 | Bengaluru |
| John      | Smith    | johnsmith1@gmail.com | 9876543211 | Bidar     |
| Roshan    | Kumar    | roshkumar1@gmail.com | 9740116621 | Belgaum   |
| Shamanth  | Gouda    | shamgoud1@gmail.com  | 9876543213 | Ballari   |
| Prathibha | Patil    | pratipatil@gmail.com | 9632657540 | Durga     |
| Sam       | john     | samjoy@gmail.com     | 9636821401 | Harihar   |
| John      | Brew     | ajay124@gmail.com    | 9636821402 | Gokak     |
| Robert    | Downy    | john123@gmail.com    | 9636821403 | Gokarna   |
| Samson    | Rock     | samrok123@gmail.com  | 9636821405 | Coorg     |
+-----------+----------+----------------------+------------+-----------+
10 rows in set (0.00 sec)
```

4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.

```
mysql> SELECT
    ->     P.ProductName,
    ->     SUM(OD.Quantity) AS TotalQuantityOrdered
    -> FROM
    ->     OrderDetails OD
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    -> GROUP BY
    ->     P.ProductName
    -> ORDER BY
    ->     TotalQuantityOrdered DESC
    -> LIMIT 1;
+-------------+----------------------+
| ProductName | TotalQuantityOrdered |
+-------------+----------------------+
| Laptop      |                    7 |
+-------------+----------------------+
1 row in set (0.00 sec)
```

5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

```
mysql> SELECT
    ->     ProductName
    -> FROM
    ->     Products
    -> ;
+-------------+
| ProductName |
+-------------+
| Laptop      |
| Mobile      |
| Tab         |
| System      |
| VR          |
| AR          |
| Watch       |
| Earphones   |
| TV          |
| Mac         |
| Iphone      |
+-------------+
11 rows in set (0.00 sec)
```

6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.

```
mysql> SELECT
    ->     C.FirstName,
    ->     C.LastName,
    ->     AVG(O.TotalAmount) AS AverageOrderValue
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> GROUP BY
    ->     C.CustomerID, C.FirstName, C.LastName;
+-----------+----------+-------------------+
| FirstName | LastName | AverageOrderValue |
+-----------+----------+-------------------+
| Harshad   | Malhotra |      68000.000000 |
| Ram       | Goud     |      42000.000000 |
| John      | Smith    |      45000.000000 |
| Roshan    | Kumar    |      55000.000000 |
| Shamanth  | Gouda    |      32000.000000 |
| Prathibha | Patil    |      31000.000000 |
| Sam       | john     |      45000.000000 |
| John      | Brew     |      12000.000000 |
| Robert    | Downy    |      91000.000000 |
| Samson    | Rock     |      51000.000000 |
+-----------+----------+-------------------+
10 rows in set (0.00 sec)
```

7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.

```
mysql> SELECT
    ->     O.OrderID,
    ->     O.CustomerID,
    ->     C.FirstName,
    ->     C.LastName,
    ->     O.TotalAmount AS TotalRevenue
    -> FROM
    ->     Orders2 O
    -> JOIN
    ->     Customers C ON O.CustomerID = C.CustomerID
    -> ORDER BY
    ->     O.TotalAmount DESC
    -> LIMIT 1;
+---------+------------+-----------+----------+--------------+
| OrderID | CustomerID | FirstName | LastName | TotalRevenue |
+---------+------------+-----------+----------+--------------+
|       9 |        109 | Robert    | Downy    |     91000.00 |
+---------+------------+-----------+----------+--------------+
1 row in set (0.00 sec)
```

8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

```
mysql> SELECT
    ->     P.ProductName,
    ->     COUNT(OD.ProductID) AS NumberOfOrders
    -> FROM
    ->     Products P
    -> LEFT JOIN
    ->     OrderDetails OD ON P.ProductID = OD.ProductID
    -> GROUP BY
    ->     P.ProductName;
+-------------+----------------+
| ProductName | NumberOfOrders |
+-------------+----------------+
| Laptop      |              3 |
| Mobile      |              0 |
| Tab         |              2 |
| System      |              0 |
| VR          |              0 |
| AR          |              0 |
| Watch       |              1 |
| Earphones   |              0 |
| TV          |              2 |
| Mac         |              1 |
| Iphone      |              0 |
+-------------+----------------+
11 rows in set (0.01 sec)
```

9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.

```
mysql> SELECT
    ->     C.FirstName,
    ->     C.LastName,
    ->     C.Email,
    ->     C.Phone,
    ->     C.Address
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> JOIN
    ->     OrderDetails OD ON O.OrderID = OD.OrderID
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    -> WHERE
    ->     P.ProductName = @ProductName;
Empty set (0.01 sec)
```

10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

```
mysql> SELECT
    ->     SUM(O.TotalAmount) AS TotalRevenue
    -> FROM
    ->     Orders2 O
    -> WHERE
    ->     O.OrderDate BETWEEN @StartDate AND @EndDate;
+--------------+
| TotalRevenue |
+--------------+
|         NULL |
+--------------+
1 row in set (0.00 sec)
```

Task 4. Subquery and its type:

1. Write an SQL query to find out which customers have not placed any orders

```
mysql> SELECT
    ->     C.CustomerID,
    ->     C.FirstName,
    ->     C.LastName,
    ->     C.Email,
    ->     C.Phone,
    ->     C.Address
    -> FROM
    ->     Customers C
    -> LEFT JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> WHERE
    ->     O.CustomerID IS NULL;
+------------+-----------+----------+--------------------+------------+-----------+
| CustomerID | FirstName | LastName | Email              | Phone      | Address   |
+------------+-----------+----------+--------------------+------------+-----------+
|        105 | Suchetha  | Patil    | suchpatil1@gmail.com | 8762530197 | Davangere |
+------------+-----------+----------+--------------------+------------+-----------+
1 row in set (0.00 sec)
```

2. Write an SQL query to find the total number of products available for sale.

```
mysql> SELECT COUNT(*) AS TotalProducts
    -> FROM Products;
+---------------+
| TotalProducts |
+---------------+
|            11 |
+---------------+
1 row in set (0.01 sec)
```

3. Write an SQL query to calculate the total revenue generated by TechShop.

```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders2;
+---------------+
| TotalRevenue |
+---------------+
|    472000.00 |
+---------------+
1 row in set (0.00 sec)
```

4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

```
mysql> SELECT
    ->     AVG(OD.Quantity) AS AverageQuantityOrdered
    -> FROM
    ->     OrderDetails OD
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    -> ;
+------------------------+
| AverageQuantityOrdered |
+------------------------+
|                 1.7778 |
+------------------------+
1 row in set (0.00 sec)
```

5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

```
mysql> SELECT
    ->     SUM(O.TotalAmount) AS TotalRevenue
    -> FROM
    ->     Orders2 O
    -> WHERE
    ->     O.CustomerID = @CustomerID;
+--------------+
| TotalRevenue |
+--------------+
|         NULL |
+--------------+
1 row in set (0.00 sec)
```

6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.

```
mysql> SELECT
    ->     C.FirstName,
    ->     C.LastName,
    ->     COUNT(O.OrderID) AS NumberOfOrders
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> GROUP BY
    ->     C.CustomerID, C.FirstName, C.LastName
    -> ORDER BY
    ->     NumberOfOrders DESC
    -> LIMIT 1;
+-----------+----------+----------------+
| FirstName | LastName | NumberOfOrders |
+-----------+----------+----------------+
| Harshad   | Malhotra |              1 |
+-----------+----------+----------------+
1 row in set (0.00 sec)
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

```
mysql> SELECT
    ->
    ->     SUM(OD.Quantity) AS TotalQuantityOrdered
    -> FROM
    ->     OrderDetails OD
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    -> ORDER BY
    ->     TotalQuantityOrdered DESC
    -> LIMIT 1;
+----------------------+
| TotalQuantityOrdered |
+----------------------+
|                   16 |
+----------------------+
1 row in set (0.00 sec)
```

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

```
mysql> SELECT
    ->     C.FirstName,
    ->     C.LastName,
    ->     SUM(O.TotalAmount) AS TotalSpending
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> JOIN
    ->     OrderDetails OD ON O.OrderID = OD.OrderID
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    -> GROUP BY
    ->     C.CustomerID, C.FirstName, C.LastName
    -> ORDER BY
    ->     TotalSpending DESC
    -> LIMIT 1;
+-----------+----------+---------------+
| FirstName | LastName | TotalSpending |
+-----------+----------+---------------+
| Samson    | Rock     |     102000.00 |
+-----------+----------+---------------+
1 row in set (0.00 sec)
```

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers

```
mysql> SELECT
    ->     C.CustomerID,
    ->     C.FirstName,
    ->     C.LastName,
    ->     COUNT(O.OrderID) AS NumberOfOrders,
    ->     SUM(O.TotalAmount) AS TotalRevenue,
    ->     AVG(O.TotalAmount) AS AverageOrderValue
    -> FROM
    ->     Customers C
    -> LEFT JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> GROUP BY
    ->     C.CustomerID, C.FirstName, C.LastName;
+------------+-----------+----------+----------------+-------------+-------------------+
| CustomerID | FirstName | LastName | NumberOfOrders | TotalRevenue | AverageOrderValue |
+------------+-----------+----------+----------------+-------------+-------------------+
|         11 | Harshad   | Malhotra |              1 |     68000.00 |      68000.000000 |
|        101 | Ram       | Goud     |              1 |     42000.00 |      42000.000000 |
|        102 | John      | Smith    |              1 |     45000.00 |      45000.000000 |
|        103 | Roshan    | Kumar    |              1 |     55000.00 |      55000.000000 |
|        104 | Shamanth  | Gouda    |              1 |     32000.00 |      32000.000000 |
|        105 | Suchetha  | Patil    |              0 |         NULL |              NULL |
|        106 | Prathibha | Patil    |              1 |     31000.00 |      31000.000000 |
|        107 | Sam       | john     |              1 |     45000.00 |      45000.000000 |
|        108 | John      | Brew     |              1 |     12000.00 |      12000.000000 |
|        109 | Robert    | Downy    |              1 |     91000.00 |      91000.000000 |
|        110 | Samson    | Rock     |              1 |     51000.00 |      51000.000000 |
+------------+-----------+----------+----------------+-------------+-------------------+
11 rows in set (0.00 sec)
```

10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

```
mysql> SELECT
    ->     C.CustomerID,
    ->     C.FirstName,
    ->     C.LastName,
    ->     COUNT(O.OrderID) AS OrderCount
    -> FROM
    ->     Customers C
    -> LEFT JOIN
    ->     Orders2 O ON C.CustomerID = O.CustomerID
    -> GROUP BY
    ->     C.CustomerID, C.FirstName, C.LastName;
+------------+-----------+----------+------------+
| CustomerID | FirstName | LastName | OrderCount |
+------------+-----------+----------+------------+
|         11 | Harshad   | Malhotra |          1 |
|        101 | Ram       | Goud     |          1 |
|        102 | John      | Smith    |          1 |
|        103 | Roshan    | Kumar    |          1 |
|        104 | Shamanth  | Gouda    |          1 |
|        105 | Suchetha  | Patil    |          0 |
|        106 | Prathibha | Patil    |          1 |
|        107 | Sam       | john     |          1 |
|        108 | John      | Brew     |          1 |
|        109 | Robert    | Downy    |          1 |
|        110 | Samson    | Rock     |          1 |
+------------+-----------+----------+------------+
11 rows in set (0.00 sec)
```